Device Recognition Authentication API Guide

Introduction

Use this guide to configure the SecureAuth Authentication API to enable end users to securely access resources using their unique device or browser profiles without requiring additional one-time passcodes (OTPs) for multi-factor authentication.

End users enroll profiles for mobile or desktop devices by successfully authenticating through a SecureAuth IdP realm. Device or browser profiles can be configured to store a credential in the device, thereby increasing security by having a client-side and server-side component that must match for a successful authentication.

Device or browser profiles can be revoked instantly at any time by the administrator or the end user.

Refer to Device Recognition for more information.

The Authentication API includes two options for Device Recognition endpoints, Standard and Stateless.

- **Standard** enables validation and confirmation of device or browser profiles with cached information, relying on the fingerprint ID for account storage.
- **Stateless** enables validation and confirmation without requiring cached storage or the fingerprint ID, enabling workflow on multiple appliances.

Device Recognition Workflow
The Authentication API uses two endpoints and either two or three requests for Device Recognition.

1. The first endpoint is a **GET** request to retrieve the JavaScript source file required to collect the digital profile from the end user's device or browser.

   - Desktop / browser profile collection options

     For **desktop browsers**, there are two options:

     - **No Cookie**: SecureAuth IdP collects information sent from the browser without delivering or registering any information at the client side, such as HTTP headers and cookies.
     - **Cookie**: SecureAuth IdP collects information sent from the browser and registers a cookie at the client-side to increase security.

   - Mobile Device (iOS and Android) profile collection methods

     For **iOS and Android mobile devices**, SecureAuth IdP collects information in two ways:

     - **Cookie mode**: SecureAuth IdP collects information sent from the browser and registers a cookie at the client-side to increase security.
     - **App mode**: SecureAuth's native mobile app pulls unique device hardware information, such as UDID, Advertiser ID, and Device ID.

2. After a successful multi-factor authentication, the profile is collected and stored in the user account in the directory.

3. When the end user utilizes the same device or browser to log into SecureAuth IdP again, the second endpoint is called via a **POST** request to compare the presented profile with those stored in the user account.

   - The current client-unique information (**a new device or browser profile**) is collected and compared with the previously registered profiles for authentication.

   - If a match for the profile is found and has an acceptable **Authentication Threshold** score, then the end user is not required to undergo additional multi-factor authentication (OTP).

4. If a match for the profile is not found or if an update is needed, then another **POST** request is used to create or update the profile.
Prerequisites

1. Complete the steps in the Authentication API Guide.
2. Complete the Device Recognition configuration steps on the SecureAuth IdP Web Admin for SecureAuth IdP v9.1 or greater. The configuration steps are for all realms that use Device Recognition.

SecureAuth IdP Web Admin

SecureAuth IdP Configuration Steps

Standard Endpoints

Three types of /dfp endpoints are used for Standard Device Recognition.

1. The /dfp/js endpoint uses the GET method to retrieve the JavaScript reference that is required to generate device or browser profiles.

   Using the JavaScript reference, the end user's devices or browsers are analyzed and most of the required information is collected; but the remaining characteristics must be provided by the application.

2. The /dfp/validate endpoint uses the POST method to compare the presented profile with those stored in the user account.

   Based on the information provided from the directory, SecureAuth IdP returns a response stating whether the profile is found or not found.

   If a match is made during a login attempt, then the profile is used as the second factor of authentication.

3. If a profile posted to the /dfp/validate endpoint returns a not_found or found_for_update status, then the information must be posted to the /dfp/confirm endpoint to create a new, or update an existing, profile in the user account in the directory.

   After the profile is validated, SecureAuth IdP returns a fingerprint_id, which is then posted to the confirm endpoint to create the entry.

To use these endpoints, configure the SecureAuth IdP realm for Device Recognition.

GET

<table>
<thead>
<tr>
<th>/dfp/js</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Endpoint</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/v1/dfp/js</td>
<td><a href="https://secureauth.company.com/secureauth2/api/v1/dfp/js">https://secureauth.company.com/secureauth2/api/v1/dfp/js</a></td>
</tr>
</tbody>
</table>

Required Fingerprint Elements

<table>
<thead>
<tr>
<th>Provided by JavaScript Reference</th>
<th>Provided by Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>uaBrowser</td>
<td>host_address (IP address of end user client)</td>
</tr>
<tr>
<td>• name</td>
<td></td>
</tr>
<tr>
<td>• version</td>
<td></td>
</tr>
<tr>
<td>• major</td>
<td></td>
</tr>
</tbody>
</table>
doNotTrack
plugins
canvas
webGl
adBlock
userTamperLanguage
userTamperScreenResolution
userTamperOS
userTamperBrowser
touchSupport
  • maxTouchPoints
  • touchEvent
  • touchStart
cookieSupport
fonts

Response to Request Example

{  "src": "https://company.secureauth.com/SecureAuth2/assets/scripts/api/digitalFingerprint.min.js?ver=9.1.0.102"}
}
The following example shows how to use the JavaScript reference to capture a client-side device or browser profile.

```html
<script>
$(function() {
    // invoke getFingerprint() method to capture
    // client-side info and serialize to JSON string.
    var serializedData = JSON.stringify(secureAuth.fingerprint.getAllResults());

    // assign JSON string to an input for posting
    // to web server.
    $('#results').val(serializedData);
});
</script>
```

### POST

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Endpoint</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/api/v1/dfp/validate</td>
<td><a href="https://secureauth.company.com/secureauth2/api/v1/dfp/validate">https://secureauth.company.com/secureauth2/api/v1/dfp/validate</a></td>
</tr>
</tbody>
</table>
Definitions

- **fingerprint_id**: GUID of the device or browser profile
- **fingerprint_name**: Descriptive name derived from the user_agent string
- **score**: Match score of the provided profile (out of 100.00)
- **match_score**: Minimum score allowed to be accepted as second factor authentication (configured in realm)
- **update_score**: Minimum score allowed to be accepted to update existing profile with new information (configured in realm, requires Multi-Factor Authentication for update)
- **status**: Status of the matching outcomes:
  - **found**: Profile found with acceptable match_score
  - **found_for_update**: Profile found with unacceptable match_score, but acceptable update_score and can be updated by posting to `/dfp/confirm` endpoint
  - **found_with_id_mismatch**: Profile found, but with different fingerprint_id
  - **not_found**: Profile not found and new one must be created by posting to `/dfp/confirm` endpoint
- **message**: Additional information regarding the status

POST Endpoint JSON Parameters and Response Examples

<table>
<thead>
<tr>
<th>JSON Parameters</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validate a known profile by including the <strong>fingerprint_id</strong> in the parameters</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;user_id&quot;: &quot;&lt;USER ID&gt;&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;host_address&quot;: &quot;&lt;IP ADDRESS&gt;&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;fingerprint_id&quot;: &quot;&lt;FP ID&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;fingerprint&quot;: { ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td>&quot;user_id&quot;: &quot;&lt;USER ID&gt;&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;host_address&quot;: &quot;&lt;IP ADDRESS&gt;&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;fingerprint&quot;: {</td>
<td></td>
</tr>
<tr>
<td>&quot;uaBrowser&quot;: {</td>
<td></td>
</tr>
<tr>
<td>&quot;name&quot;: &quot;&lt;NAME&gt;&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;version&quot;: &quot;&lt;VERSION #&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;major&quot;: &quot;&lt;MAJOR #&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>},</td>
<td></td>
</tr>
<tr>
<td>&quot;fingerprint_id&quot;: &quot;58f0f98642fe4354ba65a35a0d8bc6ff&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;fingerprint_name&quot;:</td>
<td></td>
</tr>
<tr>
<td>&quot;Windows 7 - Firefox 41.0&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;score&quot;: &quot;0.00&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;match_score&quot;: &quot;90.00&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;update_score&quot;: &quot;89.00&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;status&quot;: &quot;not_found&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;message&quot;: &quot;&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>
Example:

```json
{
  "user_id": "bdrap",
  "host_address": "188.64.168.50",
  "fingerprint": {
    "fingerprint": {
      "uaBrowser": {
        "name": "Chrome",
        "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff",
        "fingerprint_name": "Windows 7 - Firefox 41.0",
        "score": "100.00",
        "match_score": "90.00",
        "update_score": "89.00",
        "status": "found",
        "message": ""
      },
      "availableScreenResolution": {
        "timeZone": "<LOCATION>",
        "timeZoneOffset": "<OFFSET>",
        "localStorage": <T OR F>,
        "sessionStorage": <T OR F>,
        "indexedDB": <T OR F>,
        "addBehavior": <T OR F>,
        "openDatabase": <T OR F>,
        "cpuClass": <CLASS>,
        "platform": "<PLATFORM>",
        "doNotTrack": <T OR F>,
        "plugins": "<PLUGIN LIST>",
        "canvases": "<CANVAS>",
        "webGL": <WEB GL>,
        "adBlock": <T OR F>,
        "userTamperLanguage": <T OR F>,
        "userTamperScreenResolution": <T OR F>,
        "userTamperOS": <T OR F>,
        "userTamperBrowser": <T OR F>,
        "touchSupport": {
          "maxTouchPoints": <NUMBER>,
          "touchEvent": <T OR F>,
          "touchStart": <T OR F>
        },
        "cookieSupport": <T OR F>,
        "fonts": "<FONT LIST>"
      }
    }
  }
}
```
"version": "58.0.3029.110",
"major": "58"
},
"uaString": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36",
"uaDevice": {
  "model": null,
  "type": null,
  "vendor": null
},
"uaEngine": {
  "name": "WebKit",
  "version": "537.36"
},
"uaOS": {
  "name": "Mac OS",
  "version": "10.12.5"
},
"uaCPU": {
  "architecture": null
},
"uaPlatform": "MacIntel",
"language": "en-US",
"colorDepth": 24,
"pixelRatio": 1,
"screenResolution": "1920x1080",
"availableScreenResolution": "1920x1054",
"timezone": "America/Los_Angeles",
"timezoneOffset": 420,
"localStorage": true,
"sessionStorage": true,
"indexedDb": true,
"addBehavior": false,
"openDatabase": true,
"cpuClass": null,
"platform": "MacIntel",
"doNotTrack": null,
"plugins": "Shockwave Flash.application/x-shockwave-flash::swf",
"canvas": "-226092824",
"webGl": null,
Refer to the table in the /dfp/js endpoint section for more information on the required profile components

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Endpoint</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/api/v1/dfp/confirm</td>
<td><a href="https://secureauth.company.com/secureauth2/api/v1/dfp/confirm">https://secureauth.company.com/secureauth2/api/v1/dfp/confirm</a></td>
</tr>
</tbody>
</table>
Definitions

- **fingerprint_id**: GUID of the profile
- **fingerprint_name**: Descriptive name derived from the user_agent string
- **status**: Status of the matching outcomes:
  - **verified**: New profile created in user account
  - **found**: Existing profile updated in user account
  - **not_found**: Profile not created or updated, and returned a message stating: "Could not resolve fingerprint with ID." This could occur, for example, if the fingerprint was set up on a public computer. The fingerprint would not be saved and during a confirm, would not be resolved with the ID.
- **message**: Additional information regarding the status
- **user_id**: User ID provided
### POST Endpoint JSON Parameters and Response Examples

<table>
<thead>
<tr>
<th>JSON Parameters</th>
<th>Success Response</th>
<th>Failure / Error Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;user_id&quot;: &quot;&lt;USERNAME&gt;&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;fingerprint_id&quot;: &quot;&lt;FP ID&gt;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```json
{
    "user_id": "jsmith",
    "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff",
    "fingerprint_name": "Windows 7 - Firefox 41.0",
    "status": "verified",
    "message": "Fingerprint has been confirmed."
}
```

```json
{
    "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff",
    "fingerprint_name": "Windows Vista - Firefox 41.0",
    "status": "found",
    "message": "Fingerprint exists."
}
```

```json
{
    "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff",
    "fingerprint_name": "Windows 7 - Firefox 41.0",
    "status": "not_found",
    "message": "Could not resolve fingerprint with ID '38ffee7e10047418f77935'.",
    "user_id": "jsmith"
}
```

---

### Stateless Endpoints

Three types of /dfp endpoints are used for Stateless Device Recognition.

1. The /dfp/js endpoint uses the GET method to retrieve the JavaScript reference that is required to generate device or browser profiles. See endpoint information in the Standard DFP Endpoint section above.

   Using the JavaScript reference, the end user's devices or browsers are analyzed and most of the required information is collected, but the remaining characteristics must be provided by the application.

2. The /dfp/score endpoint uses the POST method to compare the presented profile with those stored in the user account.
Based on the information provided from the directory, SecureAuth IdP returns a response stating whether the profile is found or not found.

If a match is made during a login attempt, then the profile is used as the second factor of authentication.

3. The /dfp/score endpoint uses the POST method to complete the user account profile in the directory.

To use these endpoints, configure the SecureAuth IdP realm for Device Recognition.

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Endpoint</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/api/v1/dfp/score</td>
<td><a href="https://secureauth.company.com/secureauth2/api/v1/dfp/score">https://secureauth.company.com/secureauth2/api/v1/dfp/score</a></td>
</tr>
</tbody>
</table>

**Definitions**

- **fingerprint_id**: GUID of the profile
- **fingerprint_name**: Descriptive name derived from the user_agent string
- **score**: Match score of the provided profile (out of 100.00)
- **match_score**: Minimum score allowed to be accepted as second factor authentication (configured in realm)
- **update_score**: Minimum score allowed to be accepted to update existing profile with new information (configured in realm, requires multi-factor authentication for update)
- **status**: Status of the matching outcomes:
  - **found**: Profile found with acceptable match_score
  - **found_for_update**: Profile found with unacceptable match_score, but acceptable update_score and can be updated by posting to /dfp/save endpoint
  - **found_with_id_mismatch**: Profile found, but with different fingerprint_ID
  - **not_found**: Profile not found and new one must be created by posting to /dfp/save endpoint
- **message**: Additional information regarding the status

**POST Endpoint JSON Parameters and Response Examples**

<table>
<thead>
<tr>
<th>JSON Parameters</th>
<th>Response</th>
</tr>
</thead>
</table>
Score a known profile by including the `fingerprint_id` in the parameters

```json
{
    "user_id": "<USER ID>",
    "host_address": "<IP ADDRESS>",
    "fingerprint_id": "<FP ID>",
    "fingerprint": { ...
}
}
```
Example:

```json
{
    "user_id": "<USER ID>",
    "host_address": "<IP ADDRESS>",
    "fingerprint": {
        "fingerprint": {
            "uaBrowser": {
                "name": "<NAME>",
                "version": "<VERSION \\
                "major": "<MAJOR \\

            },
            "uaString": "<STRING>",
            "uaDevice": {
                "model": null,
                "type": null,
                "vendor": null
            },
            "uaEngine": {
                "name": "<NAME>",
                "version": "<VERSION \\

            },
            "uaOS": {
                "name": "<NAME>",
                "version": "<VERSION \\

            },
            "uaCPU": {
                "architecture": null
            },
            "uaPlatform": "<PLATFORM>",
            "language": "<LANGUAGE>",
            "colorDepth": "<DEPTH>",
            "pixelRatio": "<RATIO>",
            "screenResolution": "<RESOLUTION>",
            "availableScreenResolution": "<RESOLUTION>",
            "timezone": "<LOCATION>",
            "timezoneOffset": "<OFFSET>",
            "localStorage": "<T OR F>",
            "sessionStorage": "<T OR F>",
            "indexedDb": "<T OR F>",
            "addBehavior": "<T OR F>",
            "openDatabase": "<T OR F>",
            "cpuClass": "<CLASS>",
            "platform": "<PLATFORM>",
            "doNotTrack": "<T OR F>",
            "plugins": "<PLUGIN LIST>",
            "canvas": "<CANVAS>",
            "webGl": "<WEB GL>",
            "adbLock": "<T OR F>",
            "userTamperLanguage": "<T OR F>",
            "userTamperScreenResolution": "<T OR F>",
            "userTamperOS": "<T OR F>",
            "userTamperBrowser": "<T OR F>",
            "touchSupport": {
                "maxTouchPoints": null
            },
            "touchEvent": "<T OR F>",
            "touchStart": "<T OR F>",
            "cookieSupport": "<T OR F>",
            "fonts": "<FONT LIST>"
        }
    }
}
```

Example:

```json
{
    "fingerprint_id": "a31332450f284e9bbb1572e7c1c4927a",
    "fingerprint_name": "Windows 7 - Firefox 41.0",
    "score": "100.00",
    "match_score": "90.00",
    "update_score": "89.00",
    "status": "found_with_id_mismatch",
    "message": ""
}
Refer to the table in the /dfp/js endpoint section for more information on the required profile components.

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Endpoint</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/api/v1/dfp/save</td>
<td><a href="https://secureauth.company.com/secureauth2/api/v1/dfp/save">https://secureauth.company.com/secureauth2/api/v1/dfp/save</a></td>
</tr>
</tbody>
</table>
Definitions

- **fingerprint_id**: GUID of the profile
- **fingerprint_name**: Descriptive name derived from the user_agent string
- **status**: Status of the matching outcomes:
  - **found**: Matching /dfp record was found and will be replaced or updated
  - **not_found**: Existing /dfp record was not found and new one will be created
- **message**: Message shows the same response as the message for /dfp/score
- **user_id**: User ID provided

**POST Endpoint JSON Parameters and Response Examples**

<table>
<thead>
<tr>
<th>JSON Parameters</th>
<th>Response</th>
</tr>
</thead>
</table>
| ```
{ 
  "user_id": "<USER ID>",
  "host_address": "<IP ADDRESS>",
  "fingerprint_id": "<FP ID>",
  "fingerprint": 
    
    "uaBrowser": {
      "name": "<NAME>",
      "version": "<VERSION #>",
      "major": "<MAJOR #>",
    },
    
    "uaString": "<STRING>",
    "uaDevice": {
      
      "model": null,
      "type": null,
      "vendor": null
    },
    
    "uaEngine": {
      "name": "<NAME>",
      "version": "<VERSION #>",
    },
    
    "uaOS": {
      "name": "<NAME>",
      "version": "<VERSION #>",
    },
    
    "uaCPU": {
      "architecture": null
    },
    
    "uaPlatform": "<PLATFORM>",
    "language": "<LANGUAGE>",
    "colorDepth": "<DEPTH>",
    "pixelRatio": "<RATIO>",
    "screenResolution": "<RESOLUTION>",
    "availableScreenResolution":
    "timezone": "<LOCATION>",
    "timezoneOffset": "<OFFSET>",
    "localStorage": <T OR F>,
 }``` |
| ```

  
  "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff",
  
  "fingerprint_name": "Windows 7 - Firefox 41.0",
  
  "status": "not_found",
  
  "message": "",
  
  "user_id": "jsmith"

} ``` |

| | ```

{ 
  "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff",
  
  "fingerprint_name": "Windows Vista - Firefox 41.0",
  
  "status": "found",
  
  "message": "",
  
  "user_id": "jsmith"

} ``` |
"sessionStorage": <T OR F>,
"indexedDb": <T OR F>,
"addBehavior": <T OR F>,
"openDatabase": <T OR F>,
"cpuClass": <CLASS>,
"platform": "<PLATFORM>",
"doNotTrack": <T OR F>,
"plugins": "<PLUGIN LIST>",
"canvas": "<CANVAS>",
"webGl": <WEB GL>,
"adBlock": <T OR F>,
"userTamperLanguage": <T OR F>,
"userTamperScreenResolution":
<T OR F>,
"userTamperOS": <T OR F>,
"userTamperBrowser": <T OR F>,
"touchSupport": {
  "maxTouchPoints": <NUMBER>,
  "touchEvent": <T OR F>,
  "touchStart": <T OR F>
},
"cookieSupport": <T OR F>,
"fonts": "<FONT LIST>"
}

Example:
{
  "user_id": "jsmith",
  "host_address": "123.45.67.89",
  "fingerprint_id": "58f0f98642fe4354ba65a35a0d8bc6ff"
  "fingerprint": {
    "fingerprint": {
      "uaBrowser": {
        "name": "Chrome",
        "version": "58.0.3029.110",
        "major": "58"
      },
      "uaString": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36",
      "uaDevice": {
        "model": null,
        "type": null,
        "vendor": null
      },
      "uaEngine": {
        "name": "WebKit",
        "version": "537.36"
      }
    }
  }
}
"uaOS": {
  "name": "Mac OS",
  "version": "10.12.5"
},
"uaCPU": {
  "architecture": null
},
"uaPlatform": "MacIntel",
"language": "en-US",
"colorDepth": 24,
"pixelRatio": 1,
"screenResolution": "1920x1080",
"availableScreenResolution": "1920x1054",
"timezone": "America/Los_Angeles",
"timezoneOffset": 420,
"localStorage": true,
"sessionStorage": true,
"indexedDb": true,
"addBehavior": false,
"openDatabase": true,
"cpuClass": null,
"platform": "MacIntel",
"doNotTrack": null,
"plugins": "Shockwave Flash.application/x-shockwave-flash::swf",
"canvas": "-226092824",
"webGl": null,
"adBlock": false,
"userTamperLanguage": false,
"userTamperScreenResolution": false,
"userTamperOS": false,
"userTamperBrowser": false,
"touchSupport": {
  "maxTouchPoints": 0,
  "touchEvent": false,
  "touchStart": false
},
"cookieSupport": true,
"fonts": "American Typewriter,Andale Mono"}
Refer to the table in the /dfp/js endpoint section for more information on the required profile components.